Shterenlikht, A. (2019). On Quality of Implementation of Fortran 2008 Complex Intrinsic Functions on Branch Cuts. *ACM Transactions on Mathematical Software*, *45*(1), [11]. https://doi.org/10.1145/3301318

Peer reviewed version

Link to published version (if available):
10.1145/3301318

Link to publication record in Explore Bristol Research
PDF-document

This is the author accepted manuscript (AAM). The final published version (version of record) is available online via ACM at https://dl.acm.org/citation.cfm?doid=3314951.3301318 . Please refer to any applicable terms of use of the publisher.

University of Bristol - Explore Bristol Research

General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available: http://www.bristol.ac.uk/red/research-policy/pure/user-guides/ebr-terms/

Online Appendix to: On quality of implementation of Fortran 2008 complex intrinsic functions on branch cuts

A. SHTERENLIKHT, The University of Bristol, UK

ACM Reference Format:
A. Shterenlikht. 2018. Online Appendix to:, On quality of implementation of Fortran 2008 complex intrinsic functions on branch cuts. 1, 1 (November 2018), 21 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

A TEST RESULTS WITH 8 COMPILERS

Fortran compilers, compiler options and platforms used in this study are detailed in Tab. 1.

Table 1. Compilers, compiler options, operating systems and CPUs used in this work.

Compiler	Compiler options	OS	CPU
gfortran 8.2.0	`-fsign-zero`	FreeBSD	Haswell
Flang 5.0.2	`-Kieee`	FreeBSD	Haswell
Cray 8.6.5	`-hfp0`	linux	IvyBridge
Oracle 12.6 Fortran 95 8.8	`-fsimple=0 -ftrap=none`	linux	SandyBridge
PGI 18.4-0	`-Kieee`	linux	SandyBridge
Intel 18.0.3	`-assume minus0 -fp-model precise`	linux	Broadwell
NAG 6.2(Chiyoda) Build 6214	`-ieee=full`	linux	Broadwell
IBM XL Fortran 16.1.0	`-qstrict`	linux	Power8

Flang (https://github.com/flang-compiler) is an open source front end based on Nvidia PGI compiler, targeting LLVM. Flang 5.0.2 does not support inverse hyperbolic intrinsics.

Oracle 12.6 Fortran 95 compiler, released in May 2017, supports some Fortran 2003 and 2008 features, but not complex arguments for inverse trigonometric intrinsics or that on output 'a negative internal value in the field shall be prefixed with a minus sign' [2]. Hence all negative zero internal values appear in formatted output as positive zeros. This presents the appearance that the tests have failed, where the negative zero output is expected. We consider such tests as passed, but add flag "g" in the result tables below. Although adding a test for this condition is possible (need to read the printed values in and check that the signs are correct), this was outside of the main purpose of the tests.

PGI 18.4-0 does not support complex arguments to inverse hyperbolic intrinsics.

The results are presented in the form of tables. The dot entry, "·", means the test has passed. Several kinds of test failures are distinguished, as detailed in Tab. 2. Multiple failures are possible

Author's address: A. Shterenlikht, The University of Bristol, Department of Mechanical Engineering, Queen's Building, University Walk, Bristol, BS8 1TR, UK, mexas@bristol.ac.uk.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2018 Association for Computing Machinery.
XXXX-XXXX/2018/11-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

in a single test, e.g. an entry "*osz*" means that failures of kind "*o*", "*s*" and "*z*" have occurred in that test. Most failure types are self-explanatory, except type "*m*", which is justified in App. B.9.

Table 2. List of test failure types. The symbols are used in Tabs. 3, 4 and 5.

Symbol	Failure type
×	intrinsic not implemented with this argument
d	subnormal value returned, but subnormals are not supported
g	internal representation of −0 is *printed* as +0. Oracle 12.6 only
m	wrong magnitude of finite non-zero real/imaginary part
n	NaN, the correct value is finite or infinite
o	unwarranted overflow, the correct value is finite
p	normal finite non-zero result, the correct value is 0
s	wrong sign of real/imaginary part, or both
z	zero real/imaginary part, the correct value is normal finite non-zero

A.1 REAL32

For `REAL32` real and complex variables t = `TINY(0.0_REAL32)` $\approx 1.2 \times 10^{-38}$, h = `HUGE(0.0_REAL32)` $\approx 3.4 \times 10^{38}$ and e = `EPSILON(0.0_REAL32)` $\approx 1.2 \times 10^{-7}$. The test results are summarised in Tab. 3.

Table 3. Test results for `REAL32` kind.

Test	GCC	Flang	Cray	Oracle	PGI	Intel	NAG	IBM
$\log(-\infty + \mathrm{i}0)$	·	·	·	·	·	·	·	·
$\log(-h + \mathrm{i}0)$	·	·	·	·	·	·	·	·
$\log(-1 + \mathrm{i}0)$	·	·	·	·	·	·	·	·
$\log(-t + \mathrm{i}0)$	·	·	·	·	·	·	·	·
$\log(-t - \mathrm{i}0)$	·	·	·	·	·	·	·	*s*
$\log(-1 - \mathrm{i}0)$	·	·	·	·	·	·	·	*s*
$\log(-h - \mathrm{i}0)$	·	·	·	·	·	·	·	*s*
$\log(-\infty - \mathrm{i}0)$	·	·	·	·	·	·	·	*s*
$\sqrt{-\infty + \mathrm{i}0}$	·	·	·	·	·	·	·	·
$\sqrt{-h + \mathrm{i}0}$	·	*o*	·	·	·	·	·	·
$\sqrt{-1 + \mathrm{i}0}$	·	·	·	·	·	·	·	·
$\sqrt{-t + \mathrm{i}0}$	·	·	·	·	·	·	·	·
$\sqrt{-0 + \mathrm{i}0}$	·	·	·	·	·	·	·	·
$\sqrt{-0 - \mathrm{i}0}$	·	*s*	·	*g*	·	·	·	*s*
$\sqrt{-t - \mathrm{i}0}$	·	·	·	·	·	·	·	*s*
$\sqrt{-1 - \mathrm{i}0}$	·	·	·	·	·	·	·	*s*
$\sqrt{-h - \mathrm{i}0}$	·	*o*	·	·	·	·	·	*s*
$\sqrt{-\infty - \mathrm{i}0}$	·	·	*n*	·	·	·	·	*s*
$\arcsin(-\infty + \mathrm{i}0)$	·	·	·	×	·	·	*n*	·
$\arcsin(-h + \mathrm{i}0)$	·	·	*os*	×	*os*	·	·	*o*
$\arcsin(-1 + \mathrm{i}0)$	·	·	*s*	×	*s*	·	·	·
$\arcsin(-1 - \mathrm{i}0)$	·	·	·	×	·	·	·	·
$\arcsin(-h - \mathrm{i}0)$	·	·	*o*	×	*o*	·	·	*o*
$\arcsin(-\infty - \mathrm{i}0)$	·	·	·	×	·	·	*n*	·
$\arcsin(+\infty + \mathrm{i}0)$	·	·	·	×	·	·	*n*	·
$\arcsin(h + \mathrm{i}0)$	·	·	*os*	×	*os*	·	·	*o*
$\arcsin(1 + \mathrm{i}0)$	·	·	*s*	×	*s*	·	·	·
$\arcsin(1 - \mathrm{i}0)$	·	·	·	×	·	·	·	·

Table 3. Test results for REAL32 kind.

Test	GCC	Flang	Cray	Oracle	PGI	Intel	NAG	IBM
$\arcsin(h - \mathrm{i}0)$	·	·	o	×	o	·	·	o
$\arcsin(+\infty - \mathrm{i}0)$	·	·	·	×	·	·	n	·
$\arccos(-\infty + \mathrm{i}0)$	·	·	·	×	·	·	n	·
$\arccos(-h + \mathrm{i}0)$	·	·	osz	×	osz	·	·	o
$\arccos(-1 + \mathrm{i}0)$	·	·	s	×	s	·	·	·
$\arccos(-1 - \mathrm{i}0)$	·	·	·	×	·	·	·	·
$\arccos(-h - \mathrm{i}0)$	·	·	o	×	o	·	·	o
$\arccos(-\infty - \mathrm{i}0)$	·	·	·	×	·	·	n	·
$\arccos(+\infty + \mathrm{i}0)$	·	·	·	×	·	·	n	·
$\arccos(h + \mathrm{i}0)$	·	·	ops	×	ops	·	·	o
$\arccos(1 + \mathrm{i}0)$	·	·	s	×	s	·	·	·
$\arccos(1 - \mathrm{i}0)$	·	·	·	×	·	·	·	·
$\arccos(h - \mathrm{i}0)$	·	·	o	×	o	·	·	o
$\arccos(+\infty - \mathrm{i}0)$	·	·	·	×	·	·	n	·
$\arctan(+0 + \mathrm{i}\infty)$	·	n	·	×	·	·	nz	·
$\arctan(+0 + \mathrm{i}h)$	·	d	n	×	n	·	nz	n
$\arctan(+0 + \mathrm{i}(1 + e))$	·	·	·	×	·	·	nz	·
$\arctan(+0 + \mathrm{i}1)$	·	·	·	×	·	·	·	·
$\arctan(-0 + \mathrm{i}1)$	·	s	·	×	·	·	·	·
$\arctan(-0 + \mathrm{i}(1 + e))$	·	s	·	×	·	·	nz	·
$\arctan(-0 + \mathrm{i}h)$	·	ds	n	×	n	·	nz	n
$\arctan(-0 + \mathrm{i}\infty)$	·	n	·	×	·	·	nz	·
$\arctan(-0 - \mathrm{i}\infty)$	·	n	·	×	·	·	nz	·
$\arctan(-0 - \mathrm{i}h)$	·	d	n	×	n	·	nz	n
$\arctan(-0 - \mathrm{i}(1 + e))$	·	·	·	×	·	·	nz	·
$\arctan(-0 - \mathrm{i}1)$	·	·	·	×	·	·	·	·
$\arctan(+0 - \mathrm{i}1)$	·	·	·	×	·	·	·	·
$\arctan(+0 - \mathrm{i}(1 + e))$	·	·	·	×	·	·	nz	·
$\arctan(+0 - \mathrm{i}h)$	·	d	n	×	n	·	nz	n
$\arctan(+0 - \mathrm{i}\infty)$	·	n	·	×	·	·	nz	·
$\operatorname{arcsinh}(+0 + \mathrm{i}\infty)$	·	×	·	·	×	·	n	·
$\operatorname{arcsinh}(+0 + \mathrm{i}h)$	·	×	o	o	×	·	·	o
$\operatorname{arcsinh}(+0 + \mathrm{i}1)$	·	×	·	·	×	·	·	·
$\operatorname{arcsinh}(-0 + \mathrm{i}1)$	·	×	s	s	×	·	·	·
$\operatorname{arcsinh}(-0 + \mathrm{i}h)$	·	×	os	os	×	·	·	o
$\operatorname{arcsinh}(-0 + \mathrm{i}\infty)$	·	×	·	·	×	·	n	·
$\operatorname{arcsinh}(+0 - \mathrm{i}\infty)$	·	×	·	·	×	·	n	·
$\operatorname{arcsinh}(+0 - \mathrm{i}h)$	·	×	o	o	×	·	·	o
$\operatorname{arcsinh}(+0 - \mathrm{i}1)$	·	×	·	·	×	·	·	·
$\operatorname{arcsinh}(-0 - \mathrm{i}1)$	·	×	s	s	×	·	·	·
$\operatorname{arcsinh}(-0 - \mathrm{i}h)$	·	×	os	os	×	·	·	o
$\operatorname{arcsinh}(-0 - \mathrm{i}\infty)$	·	×	·	·	×	·	n	·
$\operatorname{arccosh}(-\infty + \mathrm{i}0)$	·	×	·	·	×	·	·	·
$\operatorname{arccosh}(-h + \mathrm{i}0)$	·	×	o	o	×	·	m	·
$\operatorname{arccosh}(-1 + \mathrm{i}0)$	·	×	·	·	×	·	·	·
$\operatorname{arccosh}(+0 + \mathrm{i}0)$	·	×	·	·	×	·	ps	·
$\operatorname{arccosh}(+1 + \mathrm{i}0)$	·	×	·	·	×	·	·	·
$\operatorname{arccosh}(+1 - \mathrm{i}0)$	·	×	·	g	×	·	s	·
$\operatorname{arccosh}(+0 - \mathrm{i}0)$	·	×	·	·	×	·	ps	·
$\operatorname{arccosh}(-1 - \mathrm{i}0)$	·	×	·	·	×	·	s	·
$\operatorname{arccosh}(-h - \mathrm{i}0)$	·	×	o	o	×	·	ms	·
$\operatorname{arccosh}(-\infty - \mathrm{i}0)$	·	×	·	·	×	·	s	·
$\operatorname{arctanh}(+\infty + \mathrm{i}0)$	·	×	·	·	×	·	nz	·
$\operatorname{arctanh}(h + \mathrm{i}0)$	·	×	n	n	×	·	nz	n

Table 3. Test results for REAL32 kind.

Test	GCC	Flang	Cray	Oracle	PGI	Intel	NAG	IBM
$\mathrm{arctanh}(1 + e + \mathrm{i}0)$	·	×	·	·	×	·	nz	·
$\mathrm{arctanh}(1 + \mathrm{i}0)$	·	×	·	·	×	·	·	·
$\mathrm{arctanh}(1 - \mathrm{i}0)$	·	×	·	g	×	·	·	·
$\mathrm{arctanh}(1 + e - \mathrm{i}0)$	·	×	·	·	×	·	nz	·
$\mathrm{arctanh}(h - \mathrm{i}0)$	·	×	n	n	×	·	nz	n
$\mathrm{arctanh}(+\infty - \mathrm{i}0)$	·	×	·	·	×	·	nz	·
$\mathrm{arctanh}(-\infty + \mathrm{i}0)$	·	×	·	g	×	·	nz	·
$\mathrm{arctanh}(-h + \mathrm{i}0)$	·	×	n	n	×	·	nz	n
$\mathrm{arctanh}(-1 - e + \mathrm{i}0)$	·	×	·	·	×	·	nz	·
$\mathrm{arctanh}(-1 + \mathrm{i}0)$	·	×	·	·	×	·	·	·
$\mathrm{arctanh}(-1 - \mathrm{i}0)$	·	×	·	g	×	·	·	·
$\mathrm{arctanh}(-1 - e - \mathrm{i}0)$	·	×	·	·	×	·	nz	·
$\mathrm{arctanh}(-h - \mathrm{i}0)$	·	×	n	n	×	·	nz	n
$\mathrm{arctanh}(-\infty - \mathrm{i}0)$	·	×	·	g	×	·	nz	·
Pass rate	96/96	45/58	67/96	44/56	42/58	96/96	53/96	67/96

A.2 REAL64

For REAL64 real and complex variables t = `TINY(0.0_REAL64)` $\approx 2.2 \times 10^{-308}$, h = `HUGE(0.0_REAL64)` $\approx 1.8 \times 10^{308}$ and e = `EPSILON(0.0_REAL64)` $\approx 2.2 \times 10^{-16}$. The test results are summarised in Tab. 4.

Table 4. Test results for REAL64 kind.

Test	GCC	Flang	Cray	Oracle	PGI	Intel	NAG	IBM
$\log(-\infty + \mathrm{i}0)$	·	·	·	·	·	·	·	·
$\log(-h + \mathrm{i}0)$	·	·	·	·	·	·	·	·
$\log(-1 + \mathrm{i}0)$	·	·	·	·	·	·	·	·
$\log(-t + \mathrm{i}0)$	·	·	·	·	·	·	·	·
$\log(-t - \mathrm{i}0)$	·	·	·	·	·	·	·	s
$\log(-1 - \mathrm{i}0)$	·	·	·	·	·	·	·	s
$\log(-h - \mathrm{i}0)$	·	·	·	·	·	·	·	s
$\log(-\infty - \mathrm{i}0)$	·	·	·	·	·	·	·	s
$\sqrt{-\infty + \mathrm{i}0}$	·	·	·	·	·	·	·	·
$\sqrt{-h + \mathrm{i}0}$	·	o	·	o	·	·	·	o
$\sqrt{-1 + \mathrm{i}0}$	·	·	·	·	·	·	·	·
$\sqrt{-t + \mathrm{i}0}$	·	·	·	·	·	·	·	·
$\sqrt{-0 + \mathrm{i}0}$	·	·	·	·	·	·	·	·
$\sqrt{-0 - \mathrm{i}0}$	·	s	·	g	·	·	·	s
$\sqrt{-t - \mathrm{i}0}$	·	·	·	·	·	·	·	s
$\sqrt{-1 - \mathrm{i}0}$	·	·	·	·	·	·	·	s
$\sqrt{-h - \mathrm{i}0}$	·	o	o	o	·	·	·	os
$\sqrt{-\infty - \mathrm{i}0}$	·	·	n	·	·	·	·	s
$\arcsin(-\infty + \mathrm{i}0)$	·	·	·	×	·	·	n	·
$\arcsin(-h + \mathrm{i}0)$	·	·	os	×	os	·	·	o
$\arcsin(-1 + \mathrm{i}0)$	·	·	s	×	s	·	·	·
$\arcsin(-1 - \mathrm{i}0)$	·	·	·	×	·	·	·	·
$\arcsin(-h - \mathrm{i}0)$	·	·	o	×	o	·	·	o
$\arcsin(-\infty - \mathrm{i}0)$	·	·	·	×	·	·	n	·
$\arcsin(+\infty + \mathrm{i}0)$	·	·	·	×	·	·	n	·
$\arcsin(h + \mathrm{i}0)$	·	·	os	×	os	·	·	o
$\arcsin(1 + \mathrm{i}0)$	·	·	s	×	s	·	·	·
$\arcsin(1 - \mathrm{i}0)$	·	·	·	×	·	·	·	·

Table 4. Test results for REAL64 kind.

Test	GCC	Flang	Cray	Oracle	PGI	Intel	NAG	IBM
$\arcsin(h - i0)$	·	·	*o*	×	*o*	·	·	*o*
$\arcsin(+\infty - i0)$	·	·	·	×	·	·	*n*	·
$\arccos(-\infty + i0)$	·	·	·	×	·	·	*n*	·
$\arccos(-h + i0)$	·	·	*osz*	×	*osz*	·	·	*o*
$\arccos(-1 + i0)$	·	·	*s*	×	*s*	·	·	·
$\arccos(-1 - i0)$	·	·	·	×	·	·	·	·
$\arccos(-h - i0)$	·	·	*o*	×	*o*	·	·	*o*
$\arccos(-\infty - i0)$	·	·	·	×	·	·	*n*	·
$\arccos(+\infty + i0)$	·	·	·	×	·	·	*n*	·
$\arccos(h + i0)$	·	·	*ops*	×	*ops*	·	·	*o*
$\arccos(1 + i0)$	·	·	*s*	×	*s*	·	·	·
$\arccos(1 - i0)$	·	·	·	×	·	·	·	·
$\arccos(h - i0)$	·	·	*o*	×	*o*	·	·	*o*
$\arccos(+\infty - i0)$	·	·	·	×	·	·	*n*	·
$\arctan(+0 + i\infty)$	·	*n*	·	×	·	·	*nz*	·
$\arctan(+0 + ih)$	·	*d*	*n*	×	*n*	·	*nz*	*n*
$\arctan(+0 + i(1 + e))$	·	·	·	×	·	·	*nz*	·
$\arctan(+0 + i1)$	·	·	·	×	·	·	·	·
$\arctan(-0 + i1)$	·	*s*	·	×	·	·	·	·
$\arctan(-0 + i(1 + e))$	·	*s*	·	×	·	·	*nz*	·
$\arctan(-0 + ih)$	·	*ds*	*n*	×	*n*	·	*nz*	*n*
$\arctan(-0 + i\infty)$	·	*n*	·	×	·	·	*nz*	·
$\arctan(-0 - i\infty)$	·	*n*	·	×	·	·	*nz*	·
$\arctan(-0 - ih)$	·	*d*	*n*	×	*n*	·	*nz*	*n*
$\arctan(-0 - i(1 + e))$	·	·	·	×	·	·	*nz*	·
$\arctan(-0 - i1)$	·	·	·	×	·	·	·	·
$\arctan(+0 - i1)$	·	·	·	×	·	·	·	·
$\arctan(+0 - i(1 + e))$	·	·	·	×	·	·	*nz*	·
$\arctan(+0 - ih)$	·	*d*	*n*	×	*n*	·	*nz*	*n*
$\arctan(+0 - i\infty)$	·	*n*	·	×	·	·	*nz*	·
$\text{arcsinh}(+0 + i\infty)$	·	×	·	·	×	·	*n*	·
$\text{arcsinh}(+0 + ih)$	·	×	*o*	*o*	×	·	·	*o*
$\text{arcsinh}(+0 + i1)$	·	×	·	·	×	·	·	·
$\text{arcsinh}(-0 + i1)$	·	×	*s*	*s*	×	·	·	·
$\text{arcsinh}(-0 + ih)$	·	×	*os*	*os*	×	·	·	*o*
$\text{arcsinh}(-0 + i\infty)$	·	×	·	·	×	·	*n*	·
$\text{arcsinh}(+0 - i\infty)$	·	×	·	·	×	·	*n*	·
$\text{arcsinh}(+0 - ih)$	·	×	*o*	*o*	×	·	·	*o*
$\text{arcsinh}(+0 - i1)$	·	×	·	·	×	·	·	·
$\text{arcsinh}(-0 - i1)$	·	×	*s*	*s*	×	·	·	·
$\text{arcsinh}(-0 - ih)$	·	×	*os*	*os*	×	·	·	*o*
$\text{arcsinh}(-0 - i\infty)$	·	×	·	·	×	·	*n*	·
$\text{arccosh}(-\infty + i0)$	·	×	·	·	×	·	*n*	·
$\text{arccosh}(-h + i0)$	·	×	*no*	*no*	×	·	*no*	·
$\text{arccosh}(-1 + i0)$	·	×	·	·	×	·	·	·
$\text{arccosh}(+0 + i0)$	·	×	·	·	×	·	·	·
$\text{arccosh}(1 + i0)$	·	×	·	·	×	·	·	·
$\text{arccosh}(1 - i0)$	·	×	·	*g*	×	·	*s*	·
$\text{arccosh}(+0 - i0)$	·	×	·	·	×	·	*s*	·
$\text{arccosh}(-1 - i0)$	·	×	·	·	×	·	*s*	·
$\text{arccosh}(-h - i0)$	·	×	*no*	*no*	×	·	*no*	·
$\text{arccosh}(-\infty - i0)$	·	×	·	·	×	·	*n*	·
$\text{arctanh}(+\infty + i0)$	·	×	·	·	×	·	*nz*	·
$\text{arctanh}(h + i0)$	·	×	*n*	*n*	×	·	*nz*	*n*

Table 4. Test results for REAL64 kind.

Test	GCC	Flang	Cray	Oracle	PGI	Intel	NAG	IBM
$\operatorname{arctanh}(1+e+\mathrm{i}0)$	·	×	·	·	×	·	nz	·
$\operatorname{arctanh}(1+\mathrm{i}0)$	·	×	·	·	×	·	·	·
$\operatorname{arctanh}(1-\mathrm{i}0)$	·	×	·	g	×	·	·	·
$\operatorname{arctanh}(1+e-\mathrm{i}0)$	·	×	·	·	×	·	nz	·
$\operatorname{arctanh}(h-\mathrm{i}0)$	·	×	n	n	×	·	nz	n
$\operatorname{arctanh}(+\infty-\mathrm{i}0)$	·	×	·	·	×	·	nz	·
$\operatorname{arctanh}(-\infty+\mathrm{i}0)$	·	×	·	g	×	·	nz	·
$\operatorname{arctanh}(-h+\mathrm{i}0)$	·	×	n	n	×	·	nz	n
$\operatorname{arctanh}(-1-e+\mathrm{i}0)$	·	×	·	·	×	·	nz	·
$\operatorname{arctanh}(-1+\mathrm{i}0)$	·	×	·	·	×	·	·	·
$\operatorname{arctanh}(-1-\mathrm{i}0)$	·	×	·	g	×	·	·	·
$\operatorname{arctanh}(-1-e-\mathrm{i}0)$	·	×	·	·	×	·	nz	·
$\operatorname{arctanh}(-h-\mathrm{i}0)$	·	×	n	n	×	·	nz	n
$\operatorname{arctanh}(-\infty-\mathrm{i}0)$	·	×	·	g	×	·	nz	·
Pass rate	96/96	45/58	66/96	42/56	42/58	96/96	53/96	66/96

A.3 REAL128

Flang 5.0 and PGI 16.3-0 do not support the REAL128 real kind, or any other extended precision kind.

In addition to the limitations mentioned in the beginning of App. A, Oracle 12.6 does not support inverse hyperbolic intrinsics for complex arguments of REAL128 kind.

No tests could be done with NAG or IBM compilers for the REAL128 kind, because although both compilers support it, their REAL128 kinds do not conform to IEEE binary128 format, i.e. the result value of `IEEE_SUPPORT_DATATYPE(1.0_REAL128)` in both NAG and IBM is false.

For REAL128 real and complex variables t = `TINY(0.0_REAL128)` $\approx 3.3\times10^{-4932}$, h = `HUGE(0.0_REAL128)` $\approx 1.2\times10^{4932}$ and e = `EPSILON(0.0_REAL128)` $\approx 1.9\times10^{-34}$. The test results are summarised in Tab. 5.

Table 5. Test results for REAL128 kind.

Test	GCC	Cray	Oracle	Intel
$\log(-\infty+\mathrm{i}0)$	·	·	·	·
$\log(-h+\mathrm{i}0)$	·	·	·	·
$\log(-1+\mathrm{i}0)$	·	·	·	·
$\log(-t+\mathrm{i}0)$	·	·	·	·
$\log(-t-\mathrm{i}0)$	·	·	·	·
$\log(-1-\mathrm{i}0)$	·	·	·	·
$\log(-h-\mathrm{i}0)$	·	·	·	·
$\log(-\infty-\mathrm{i}0)$	·	·	·	·
$\sqrt{-\infty+\mathrm{i}0}$	·	·	·	·
$\sqrt{-h+\mathrm{i}0}$	·	·	·	·
$\sqrt{-1+\mathrm{i}0}$	·	·	·	·
$\sqrt{-t+\mathrm{i}0}$	·	·	·	·
$\sqrt{-0+\mathrm{i}0}$	·	·	·	·
$\sqrt{-0-\mathrm{i}0}$	·	·	·	·
$\sqrt{-t-\mathrm{i}0}$	·	·	·	·
$\sqrt{-1-\mathrm{i}0}$	·	·	·	·
$\sqrt{-h-\mathrm{i}0}$	·	·	·	·
$\sqrt{-\infty-\mathrm{i}0}$	·	·	·	·
$\arcsin(-\infty+\mathrm{i}0)$	·	·	×	·
$\arcsin(-h+\mathrm{i}0)$	o	o	×	·

Table 5. Test results for REAL128 kind.

Test	GCC	Cray	Oracle	Intel
$\arcsin(-1+\mathrm{i}0)$	$\cdot$	$\cdot$	$\times$	$\cdot$
$\arcsin(-1-\mathrm{i}0)$	$\cdot$	$\cdot$	$\times$	$\cdot$
$\arcsin(-h-\mathrm{i}0)$	o	o	$\times$	$\cdot$
$\arcsin(-\infty-\mathrm{i}0)$	$\cdot$	$\cdot$	$\times$	$\cdot$
$\arcsin(+\infty+\mathrm{i}0)$	$\cdot$	$\cdot$	$\times$	$\cdot$
$\arcsin(h+\mathrm{i}0)$	o	o	$\times$	$\cdot$
$\arcsin(1+\mathrm{i}0)$	$\cdot$	$\cdot$	$\times$	$\cdot$
$\arcsin(1-\mathrm{i}0)$	$\cdot$	$\cdot$	$\times$	$\cdot$
$\arcsin(h-\mathrm{i}0)$	o	o	$\times$	$\cdot$
$\arcsin(+\infty-\mathrm{i}0)$	$\cdot$	$\cdot$	$\times$	$\cdot$
$\arccos(-\infty+\mathrm{i}0)$	$\cdot$	$\cdot$	$\times$	$\cdot$
$\arccos(-h+\mathrm{i}0)$	o	o	$\times$	$\cdot$
$\arccos(-1+\mathrm{i}0)$	$\cdot$	$\cdot$	$\times$	$\cdot$
$\arccos(-1-\mathrm{i}0)$	$\cdot$	$\cdot$	$\times$	$\cdot$
$\arccos(-h-\mathrm{i}0)$	o	o	$\times$	$\cdot$
$\arccos(-\infty-\mathrm{i}0)$	$\cdot$	$\cdot$	$\times$	$\cdot$
$\arccos(+\infty+\mathrm{i}0)$	$\cdot$	$\cdot$	$\times$	$\cdot$
$\arccos(h+\mathrm{i}0)$	o	o	$\times$	$\cdot$
$\arccos(1+\mathrm{i}0)$	$\cdot$	$\cdot$	$\times$	$\cdot$
$\arccos(1-\mathrm{i}0)$	$\cdot$	$\cdot$	$\times$	$\cdot$
$\arccos(h-\mathrm{i}0)$	o	o	$\times$	$\cdot$
$\arccos(+\infty-\mathrm{i}0)$	$\cdot$	$\cdot$	$\times$	$\cdot$
$\arctan(+0+\mathrm{i}\infty)$	$\cdot$	$\cdot$	$\times$	$\cdot$
$\arctan(+0+\mathrm{i}h)$	n	n	$\times$	$\cdot$
$\arctan(+0+\mathrm{i}(1+e))$	$\cdot$	$\cdot$	$\times$	$\cdot$
$\arctan(+0+\mathrm{i}1)$	$\cdot$	$\cdot$	$\times$	$\cdot$
$\arctan(-0+\mathrm{i}1)$	$\cdot$	$\cdot$	$\times$	$\cdot$
$\arctan(-0+\mathrm{i}(1+e))$	$\cdot$	$\cdot$	$\times$	$\cdot$
$\arctan(-0+\mathrm{i}h)$	n	n	$\times$	$\cdot$
$\arctan(-0+\mathrm{i}\infty)$	$\cdot$	$\cdot$	$\times$	$\cdot$
$\arctan(-0-\mathrm{i}\infty)$	$\cdot$	$\cdot$	$\times$	$\cdot$
$\arctan(-0-\mathrm{i}h)$	n	n	$\times$	$\cdot$
$\arctan(-0-\mathrm{i}(1+e))$	$\cdot$	$\cdot$	$\times$	$\cdot$
$\arctan(-0-\mathrm{i}1)$	$\cdot$	$\cdot$	$\times$	$\cdot$
$\arctan(+0-\mathrm{i}1)$	$\cdot$	$\cdot$	$\times$	$\cdot$
$\arctan(+0-\mathrm{i}(1+e))$	$\cdot$	$\cdot$	$\times$	$\cdot$
$\arctan(+0-\mathrm{i}h)$	n	n	$\times$	$\cdot$
$\arctan(+0-\mathrm{i}\infty)$	$\cdot$	$\cdot$	$\times$	$\cdot$
$\operatorname{arcsinh}(+0+\mathrm{i}\infty)$	$\cdot$	$\cdot$	$\times$	$\cdot$
$\operatorname{arcsinh}(+0+\mathrm{i}h)$	o	o	$\times$	$\cdot$
$\operatorname{arcsinh}(+0+\mathrm{i}1)$	$\cdot$	$\cdot$	$\times$	$\cdot$
$\operatorname{arcsinh}(-0+\mathrm{i}1)$	$\cdot$	$\cdot$	$\times$	$\cdot$
$\operatorname{arcsinh}(-0+\mathrm{i}h)$	o	o	$\times$	$\cdot$
$\operatorname{arcsinh}(-0+\mathrm{i}\infty)$	$\cdot$	$\cdot$	$\times$	$\cdot$
$\operatorname{arcsinh}(+0-\mathrm{i}\infty)$	$\cdot$	$\cdot$	$\times$	$\cdot$
$\operatorname{arcsinh}(+0-\mathrm{i}h)$	o	o	$\times$	$\cdot$
$\operatorname{arcsinh}(+0-\mathrm{i}1)$	$\cdot$	$\cdot$	$\times$	$\cdot$
$\operatorname{arcsinh}(-0-\mathrm{i}1)$	$\cdot$	$\cdot$	$\times$	$\cdot$
$\operatorname{arcsinh}(-0-\mathrm{i}h)$	o	o	$\times$	$\cdot$
$\operatorname{arcsinh}(-0-\mathrm{i}\infty)$	$\cdot$	$\cdot$	$\times$	$\cdot$
$\operatorname{arccosh}(-\infty+\mathrm{i}0)$	$\cdot$	$\cdot$	$\times$	$\cdot$
$\operatorname{arccosh}(-h+\mathrm{i}0)$	$\cdot$	$\cdot$	$\times$	$\cdot$
$\operatorname{arccosh}(-1+\mathrm{i}0)$	$\cdot$	$\cdot$	$\times$	$\cdot$
$\operatorname{arccosh}(+0+\mathrm{i}0)$	$\cdot$	$\cdot$	$\times$	$\cdot$

Table 5. Test results for REAL128 kind.

Test	GCC	Cray	Oracle	Intel
$\operatorname{arccosh}(1+\mathrm{i}0)$	·	·	×	·
$\operatorname{arccosh}(1-\mathrm{i}0)$	·	·	×	·
$\operatorname{arccosh}(+0-\mathrm{i}0)$	·	·	×	·
$\operatorname{arccosh}(-1-\mathrm{i}0)$	·	·	×	·
$\operatorname{arccosh}(-h-\mathrm{i}0)$	·	·	×	·
$\operatorname{arccosh}(-\infty-\mathrm{i}0)$	·	·	×	·
$\operatorname{arctanh}(+\infty+\mathrm{i}0)$	·	·	×	·
$\operatorname{arctanh}(h+\mathrm{i}0)$	n	n	×	·
$\operatorname{arctanh}(1+e+\mathrm{i}0)$	·	·	×	·
$\operatorname{arctanh}(1+\mathrm{i}0)$	·	·	×	·
$\operatorname{arctanh}(1-\mathrm{i}0)$	·	·	×	·
$\operatorname{arctanh}(1+e-\mathrm{i}0)$	·	·	×	·
$\operatorname{arctanh}(h-\mathrm{i}0)$	n	n	×	·
$\operatorname{arctanh}(+\infty-\mathrm{i}0)$	·	·	×	·
$\operatorname{arctanh}(-\infty+\mathrm{i}0)$	·	·	×	·
$\operatorname{arctanh}(-h+\mathrm{i}0)$	n	n	×	·
$\operatorname{arctanh}(-1-e+\mathrm{i}0)$	·	·	×	·
$\operatorname{arctanh}(-1+\mathrm{i}0)$	·	·	×	·
$\operatorname{arctanh}(-1-\mathrm{i}0)$	·	·	×	·
$\operatorname{arctanh}(-1-e-\mathrm{i}0)$	·	·	×	·
$\operatorname{arctanh}(-h-\mathrm{i}0)$	n	n	×	·
$\operatorname{arctanh}(-\infty-\mathrm{i}0)$	·	·	×	·
Pass rate	76/96	76/96	18/18	96/96

B ANALYTIC SOLUTIONS FOR 8 ELEMENTARY COMPLEX FUNCTIONS ON BRANCH CUTS

This section contains brief but complete derivations for the values of the 8 elementary complex functions studied in this work on branch cuts. The reader is referred to NIST Digital Library of Mathematical Functions (DLMF) [1] for all definitions. The detailed derivations are given at https://cmplx.sourceforge.io.

The polar form, for $z \neq 0$:

$$z = |z| \exp \operatorname{Arg} z \tag{1}$$

where $\operatorname{Arg} z$ is defined in Tab. 6 and ω is defined as follows (see Eqns. 1.9.5 and 1.9.6 in [1]):

$$\omega = \arctan \left| \frac{y}{x} \right|; \qquad \omega \in [0, \pi/2] \tag{2}$$

Table 6. Definition of Arg$z$.

quadrant	x	y	Arg$z$
1st	$\geq 0; +0$	$\geq 0; +0$	ω
2nd	$\leq 0; -0$	$\geq 0; +0$	$\pi - \omega$
3rd	$\leq 0; -0$	$\leq 0; -0$	$-\pi + \omega$
4th	$\geq 0; +0$	$\leq 0; -0$	$-\omega$

B.1 Base e log

From Eqn. 4.2.3 [1]:

$$\log z = \log|z| + \mathrm{i}\mathrm{Arg}z \tag{3}$$

$\log z$ has a single branch cut along the negative real axis, $x \le 0$, see Fig. 4.2.1 [1].

$z = -a - \mathrm{i}0, a > 0$ is in the 3rd quadrant, with $\omega = 0 \Rightarrow \mathrm{Arg}z = -\pi$. $|z| = a \Rightarrow \log|z| = \log a$. If $1 < a < +\infty$ then $\log|z| > 0$. If $0 < a < 1$ then $\log|z| < 0$. If $a = 1$ then $\log|z| = 0$.

$z = -a + \mathrm{i}0, a > 0$ is in the 2nd quadrant, with $\omega = 0 \Rightarrow \mathrm{Arg}z = +\pi$. The rest of the analysis follows the previous case. The results are given in Tab. 7.

Table 7. $\log z$ on the branch cut, $a > 0$.

z	$\log z$
$-a - \mathrm{i}0$	$\log a - \mathrm{i}\pi$
$-a + \mathrm{i}0$	$\log a + \mathrm{i}\pi$

B.2 Square root

From (1):

$$\sqrt{z} = \sqrt{|z|} \exp \frac{\mathrm{Arg}z}{2} \tag{4}$$

$\sqrt{z}$ has a single branch cut along the negative real axis, $x \le 0$, or $\mathrm{Arg}z = \pm\pi$. For $\mathrm{Arg}z = +\pi$, $\sqrt{z}$ is on the positive imaginary axis. For $\mathrm{Arg}z = -\pi$, $\sqrt{z}$ is on the negative imaginary axis, including $|z| = 0$:

$$\sqrt{\pm 0 + \mathrm{i}0} = +0 + \mathrm{i}0 \quad ; \quad \sqrt{\pm 0 - \mathrm{i}0} = +0 - \mathrm{i}0 \tag{5}$$

B.3 arcsin

From Eqn. 4.23.19 [1]:

$$\arcsin z = -\mathrm{i}\log(\sqrt{1-z^2} + \mathrm{i}z) \tag{6}$$

$\arcsin z$ has 2 branch cuts, see Fig. 3. Four cases are examined, one for each side of each branch cut. In all cases $a \ge 1$ is a real value.

$z = -a - \mathrm{i}0 \Rightarrow \mathrm{i}z = \mathrm{i}(-a-\mathrm{i}0) = 0 - \mathrm{i}a \Rightarrow z^2 = (-a-\mathrm{i}0)(-a-\mathrm{i}0) = a^2 + \mathrm{i}0 \Rightarrow 1 - z^2 = 1 - a^2 - \mathrm{i}0 \Rightarrow \sqrt{1-z^2} = \sqrt{1-a^2-\mathrm{i}0}$. The expression under $\sqrt{}$ is a complex number with a non-positive real part and a negative zero imaginary part. Therefore $\sqrt{1-a^2-\mathrm{i}0}$ lies on the negative imaginary axis. This can be expressed as follows: $\sqrt{1-z^2} = 0 - \mathrm{i}\sqrt{a^2-1} \Rightarrow \sqrt{1-z^2} + \mathrm{i}z = 0 - \mathrm{i}(\sqrt{a^2-1} + a)$. The imaginary part of the last expression is ≤ 0, therefore $\mathrm{Arg}(\sqrt{1-z^2} + \mathrm{i}z) = -\pi/2 \Rightarrow \log(\sqrt{1-z^2} + \mathrm{i}z) = \log(\sqrt{a^2-1} + a) - \mathrm{i}\pi/2 \Rightarrow -\mathrm{i}\log(\sqrt{1-z^2} + \mathrm{i}z) = -\mathrm{i}\log(\sqrt{a^2-1} + a) - \pi/2 = -\pi/2 - \mathrm{i}b$, where

$$b = \log(\sqrt{a^2-1} + a) \ge 0. \tag{7}$$

Hence $\arcsin z = -\pi/2 - \mathrm{i}b$. This value lies in the 3rd quadrant.

The identities $\arcsin(-z) = -\arcsin z$, Eqn. 4.23.10 [1], and $\mathrm{conj}(\arcsin z) = \arcsin(\mathrm{conj}\, z)$, clear from Eqn. 4.24.1 [1], are used to obtain expressions for $\arcsin z$ for the other sides of the branch cuts. The results are summarised in Tab. 8.

Table 8. $\arcsin z$ on branch cuts. $a \geq 1$ and b is given in Eqn. (7).

z	$\arcsin z$
$-a - \mathrm{i}0$	$-\pi/2 - \mathrm{i}b$
$-a + \mathrm{i}0$	$-\pi/2 + \mathrm{i}b$
$a - \mathrm{i}0$	$\pi/2 - \mathrm{i}b$
$a + \mathrm{i}0$	$\pi/2 + \mathrm{i}b$

B.4 arccos

From Eqns. 4.23.19 and 4.23.22 [1] $\arccos z = \pi/2 - \arcsin z$. $\arccos z$ has the same 2 branch cuts as $\arcsin z$, hence we use the same 4 expressions for z as in Sec. B.3. a and b are also as defined in Sec. B.3.

$z = -a - \mathrm{i}0 \Rightarrow \arcsin z = -\pi/2 - \mathrm{i}b \Rightarrow \arccos z = \pi + \mathrm{i}b$. This value is in the 1st quadrant.

$z = -a + \mathrm{i}0 \Rightarrow \arcsin z = -\pi/2 + \mathrm{i}b \Rightarrow \arccos z = \pi - \mathrm{i}b$. This value is in the 4th quadrant.

Using the identity $\arccos(-z) = \pi - \arccos z$ from Eqn. 4.23.11 [1] the values on the other sides of the branch cuts are obtained. The results are summarised in Tab. 9.

Table 9. $\arccos z$ on the branch cuts. $a \geq 1$ and b is given in Eqn. (7).

z	$\arccos z$
$-a - \mathrm{i}0$	$\pi + \mathrm{i}b$
$-a + \mathrm{i}0$	$\pi - \mathrm{i}b$
$a - \mathrm{i}0$	$+0 + \mathrm{i}b$
$a + \mathrm{i}0$	$+0 - \mathrm{i}b$

B.5 arctan

$\arctan z$ has 2 branch cuts along the imaginary axis, $y \geq 1$ and $y \leq -1$, See Fig. 5 and Fig. 4.23.1(ii) in [1]. Eqn. 4.23.26 in [1] has branch cuts along the real axis. Eqn. 4.23.27 [1] can be used only on branch cuts. Hence we prefer to use the identity from [3]:

$$\arctan z = \frac{\operatorname{arctanh}(\mathrm{i}z)}{\mathrm{i}} \tag{8}$$

which leads to:

$$\arctan z = \frac{\mathrm{i}}{2} \log \frac{1 - \mathrm{i}z}{1 + \mathrm{i}z} = \frac{\mathrm{i}}{2}(\log(1 - \mathrm{i}z) - \log(1 + \mathrm{i}z)) \tag{9}$$

which has the branch cuts along the imaginary axis. Both sides of both branch cuts are analysed. In the following $a \geq 1$ is a real number.

The 1st quadrant: $z = 0 + \mathrm{i}a \Rightarrow \mathrm{i}z = -a + \mathrm{i}0 \Rightarrow 1 - \mathrm{i}z = a + 1 - \mathrm{i}0 \Rightarrow \log(1 - \mathrm{i}z) = \log(a + 1 - \mathrm{i}0)$. $\operatorname{Arg}(a + 1 - \mathrm{i}0) = -0 \Rightarrow \log(1 - \mathrm{i}z) = \log(a + 1) - \mathrm{i}0$. Similarly $1 + \mathrm{i}z = 1 - a + \mathrm{i}0 \Rightarrow \log(1 + \mathrm{i}z) = \log(1 - a + \mathrm{i}0)$. $\operatorname{Arg}(1 - a + \mathrm{i}0) = \pi \Rightarrow \log(1 + \mathrm{i}z) = \log(a - 1) + \mathrm{i}\pi \Rightarrow \log(1 - \mathrm{i}z) - \log(1 + \mathrm{i}z) = \log(a + 1) - \mathrm{i}0 - \log(a - 1) - \mathrm{i}\pi = \log(a + 1)/(a - 1) - \mathrm{i}\pi$. Finally

$$\arctan z = \frac{\mathrm{i}}{2}\left(\log \frac{a + 1}{a - 1} - \mathrm{i}\pi\right) = \frac{\pi}{2} + \mathrm{i}c \tag{10}$$

where

$$c = \frac{1}{2} \log \frac{a + 1}{a - 1} \geq 0. \tag{11}$$

If $a = 1 \Rightarrow \Im(\arctan z) = +\infty$. If $a \to +\infty \Rightarrow \Im(\arctan z) \to +0$.

The identities $\arctan(-z) = -\arctan z$, Eqn. 4.23.12 [1], and $\mathrm{conj}(\arctan z) = \arctan(\mathrm{conj}\, z)$, clear from Eqn. 4.24.3 [1], are used to obtain expressions for $\arctan z$ for the values on the other sides of the branch cuts. The results are summarised in Tab. 10.

Table 10. $\arctan z$ on the branch cuts. $a \geq 1$ and c is given in Eqn. (11).

z	$\arctan z$
$+0 + \mathrm{i}a$	$\pi/2 + \mathrm{i}c$
$-0 + \mathrm{i}a$	$-\pi/2 + \mathrm{i}c$
$-0 - \mathrm{i}a$	$-\pi/2 - \mathrm{i}c$
$+0 - \mathrm{i}a$	$\pi/2 - \mathrm{i}c$

B.6 arcsinh

The most convenient expression for $\operatorname{arcsinh} z$ is:

$$\operatorname{arcsinh} z = \mathrm{i} \arcsin(-\mathrm{i}z) \tag{12}$$

which can be obtained from Table 1 in [3] or by combining Eqn. 4.37.16 [1] with Eqn. (6). Accordingly the branch cuts are moved from the real axis for $\arcsin z$ to the imaginary axis for $\operatorname{arcsinh} z$. In the following a and b are as in Sec. B.3.

$z = -0 - \mathrm{i}a \Rightarrow -\mathrm{i}z = -a + \mathrm{i}0$. From Tab. 8 $\arcsin(-a + \mathrm{i}0) = -\pi/2 + ib \Rightarrow \operatorname{arcsinh} z = -b - \mathrm{i}\pi/2$.

$z = +0 - \mathrm{i}a \Rightarrow -\mathrm{i}z = -a - \mathrm{i}0$. From Tab. 8 $\arcsin(-a - \mathrm{i}0) = -\pi/2 - ib \Rightarrow \operatorname{arcsinh} z = b - \mathrm{i}\pi/2$.

For the values on the other 2 sides of the branch cuts we use fact that arcsinh is an odd function: Eqn. 4.37.10 [1]. The results are summarised in Tab. 11.

Table 11. $\operatorname{arcsinh} z$ on branch cuts. $a \geq 1$ and b is given in Eqn. (7).

z	$\operatorname{arcsinh} z$
$+0 + \mathrm{i}a$	$b + \mathrm{i}\pi/2$
$-0 + \mathrm{i}a$	$-b + \mathrm{i}\pi/2$
$+0 - \mathrm{i}a$	$b - \mathrm{i}\pi/2$
$-0 - \mathrm{i}a$	$-b - \mathrm{i}\pi/2$

B.7 arccosh

From Table 1 in [3], which is reproduced in Eqn. 4.37.21 [1]:

$$\operatorname{arccosh} z = 2 \log\left(\sqrt{(z+1)/2} + \sqrt{(z-1)/2}\right) \tag{13}$$

$\operatorname{arccosh} z$ has a single branch cut along the real axis at $x \leq 1$.

$z = -a + \mathrm{i}0, a \geq 1 \Rightarrow \sqrt{(z+1)/2} = \sqrt{(-a+1+\mathrm{i}0)/2}$ and $\sqrt{(z-1)/2} = \sqrt{(-a-1+\mathrm{i}0)/2}$. The real parts of both expressions under $\sqrt{}$ are ≤ 0. The imaginary parts of both expressions under $\sqrt{}$ are $+0$, i.e. the Arg of both expressions under $\sqrt{}$ are $+\pi$. Hence the principal values of both square roots are on the positive imaginary axis: $\sqrt{(z+1)/2} = 0 + \mathrm{i}\sqrt{(a-1)/2}$ and $\sqrt{(z-1)/2} = 0 + \mathrm{i}\sqrt{(a+1)/2} \Rightarrow \sqrt{(z+1)/2} + \sqrt{(z-1)/2} = 0 + \mathrm{i}(\sqrt{(a-1)/2} + \sqrt{(a+1)/2})$. The imaginary part of the last expression is ≥ 1, therefore it is in the 1st quadrant. Hence $\log(\sqrt{(z+1)/2} + \sqrt{(z-1)/2}) = \log(\sqrt{(a-1)/2} +$

$\sqrt{(a+1)/2}) + \mathrm{i}\pi/2$. Further, $2\log(\sqrt{(a-1)/2} + \sqrt{(a+1)/2}) = b \Rightarrow \operatorname{arccosh} z = b + \mathrm{i}\pi$, where b is given in Eqn. (7)

$z = a + \mathrm{i}0, -1 \leq a \leq 1 \Rightarrow \sqrt{(z+1)/2} = \sqrt{(a+1+\mathrm{i}0)/2} = \sqrt{(a+1)/2} + \mathrm{i}0$. However in $\sqrt{(z-1)/2} = \sqrt{(a-1+\mathrm{i}0)/2}$ the real and the imaginary parts of the expression under $\sqrt{}$ are ≤ 0 and $+0$ respectively, meaning that the Arg of this expression is $+\pi$. Hence the principal value of $\sqrt{\ldots}$ is on the positive imaginary axis: $\sqrt{(z-1)/2} = 0 + \mathrm{i}\sqrt{(-a+1)/2} \Rightarrow \sqrt{(z+1)/2} + \sqrt{(z-1)/2} = \sqrt{(a+1)/2} + \mathrm{i}\sqrt{(-a+1)/2}$. The absolute value of this expression is 1 and $\operatorname{Arg}(\sqrt{(z+1)/2} + \sqrt{(z-1)/2}) = \arctan\sqrt{(-a+1)/(a+1)}$. Thus

$$\log\left(\sqrt{\frac{z+1}{2}} + \sqrt{\frac{z-1}{2}}\right) = 0 + \mathrm{i}\arctan\sqrt{\frac{-a+1}{a+1}} \tag{14}$$

or $\operatorname{arccosh} z = 0 + \mathrm{i}d$ where

$$d = 2\arctan\sqrt{\frac{-a+1}{a+1}} \quad ; \quad 0 \leq d \leq \pi \tag{15}$$

As $a \to -1 \Rightarrow \Im(\operatorname{arccosh} z) \to \pi$. As $a \to 1 \Rightarrow \Im(\operatorname{arccosh} z) \to 0$. If $a = 0 \Rightarrow \Im(\operatorname{arccosh} z) = \pi/2$.

Using the identity $\operatorname{conj}(\operatorname{arccosh} z) = \operatorname{arccosh}(\operatorname{conj} z)$, clear from Eqn. 4.38.3 [1], the values of $\operatorname{arccosh} z$ on the other sides of the branch cut are obtained. The results are summarised in Tab. 12.

Table 12. $\operatorname{arccosh} z$ on the branch cut. b and d are given in Eqns. (7) and (15).

z	a	$\operatorname{arccosh} z$
$-a+\mathrm{i}0$	$a \geq 1$	$b+\mathrm{i}\pi$
$a+\mathrm{i}0$	$-1 \leq a \leq 1$	$+0+\mathrm{i}d$
$a-\mathrm{i}0$	$-1 \leq a \leq 1$	$+0-\mathrm{i}d$
$-a-\mathrm{i}0$	$a \geq 1$	$b-\mathrm{i}\pi$

B.8 arctanh

From Eqn. (8) $\operatorname{arctanh} z = \mathrm{i}\arctan(-\mathrm{i}z)$. $\operatorname{arctanh} z$ has 2 branch cuts along the real axis: $x \geq 1$ and $x \leq -1$. $z = a + \mathrm{i}0, a \geq 1 \Rightarrow -\mathrm{i}z = +0 - \mathrm{i}a$. From Tab. 10 $\arctan(+0 - \mathrm{i}a) = \pi/2 - \mathrm{i}c \Rightarrow \operatorname{arctanh} z = c + \mathrm{i}\pi/2$, where c is given by Eqn. (11). The values on the other sides of the branch cuts are obtained using the fact that arctanh is an odd function, Eqn. 4.37.12 [1], and that $\operatorname{conj}(\operatorname{arctanh} z) = \operatorname{arctanh}(conjz)$, clear from Eqn. 4.38.5 [1]. The results are summarised in Tab. 13.

Table 13. $\operatorname{arctanh} z$ on the branch cuts. $a \geq 1$ and c is given in Eqn. (11).

z	$\operatorname{arctanh} z$
$a+\mathrm{i}0$	$c+\mathrm{i}\pi/2$
$-a+\mathrm{i}0$	$-c+\mathrm{i}\pi/2$
$-a-\mathrm{i}0$	$-c-\mathrm{i}\pi/2$
$a-\mathrm{i}0$	$c-\mathrm{i}\pi/2$

B.9 $\log(2h)$

Tabs. 8, 9, 11 and 12 show that on the branch cuts $\Im \arcsin z = \pm b$, $\Im \arccos z = \pm b$, $\Re \operatorname{arcsinh} z = \pm b$ and, on the part of the cut with $x \leq -1$, $\Re \operatorname{arccosh} z = b$, where b is given in Eqn. (7). The fact that the same expression for b appears as either a real or an imaginary part in these 4 complex functions on the branch cuts is used in the tests.

When calculation is done using IEEE floating point arithmetic, and $a = h$, then $b = \log(\sqrt{h^2 - 1} + h) = \log(2h)$, because within precision, p, of `REAL32`, `REAL64` or `REAL128` real kinds, $h+1 = h-1 = h$. A truncated value of $\log(2h)$, denoted `log2h`, is used in the tests for validating the calculated real or imaginary parts of arcsin, arccos, arcsinh and arccosh:

```
log2h=real(int(log(2.0_fk)+log(huge(0.0_fk))),kind=fk)
```

where `fk` is either `REAL32`, `REAL64` or `REAL128`.

The calculated difference $\log(2h) -$ `log2h` is much greater than the expected relative error in calculated $\log(2h)$, see Sec. 2 of the main paper. Hence tests can be constructed to require that the following is true: $\Im \operatorname{arccosh}(-h + \mathrm{i}0) >$ `log2h`, with similar tests for the other 3 functions. Failures of such tests are assigned type "m" in the main paper.

REFERENCES

[1] DLMF 2018. NIST Digital Library of Mathematical Functions. Release 1.0.19 of 2018-06-22. http://dlmf.nist.gov/ F. W. J. Olver, A. B. Olde Daalhuis, D. W. Lozier, B. I. Schneider, R. F. Boisvert, C. W. Clark, B. R. Miller and B. V. Saunders, eds.
[2] ISO/IEC 1539-1:2004. 2004. *Information technology – Programming languages – Fortran – Part 1: Base language.*
[3] W. Kahan. 1987. Branch Cuts for Complex Elementary Functions, or Much Ado About Nothing's Sign Bit. In *The State of the Art in Numerical Analysis*, A. Iserles and M. J. D. Powell (Eds.). Clarendon Press, Oxford.

C REFERENCE CONFORMAL MAPS OF THE BRANCH CUTS

Conformal maps of the branch cuts of the 8 intrinsics are given here as a graphical reference. These were calculated and plotted automatically using `REAL64` real and complex kind with gfortran 8.2 compiler. The intention is that users can easily replot these maps, using the code at https://cmplx.sourceforge.io, to verify that they agree with the reference ones in this appendix.

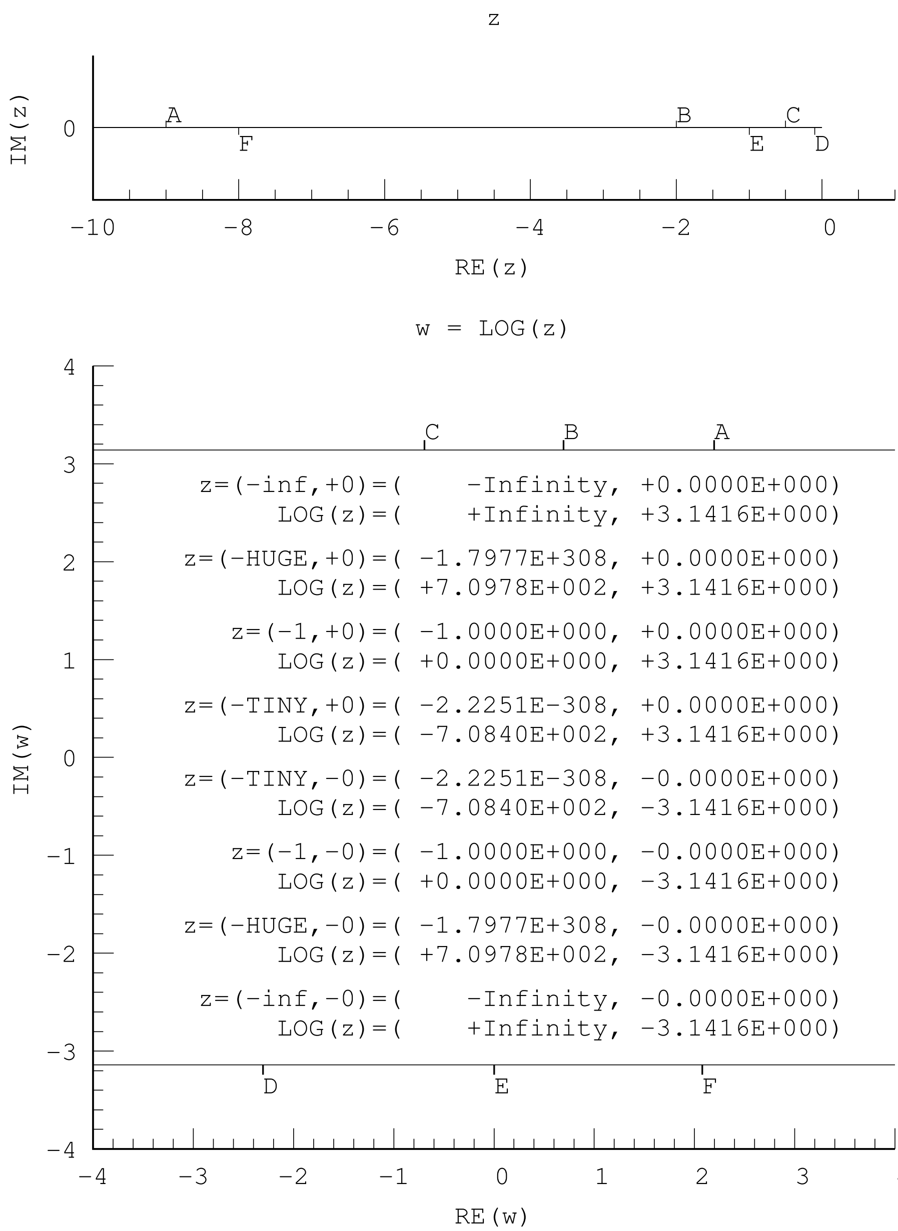

Fig. 1. Map of the branch cut of $w = \log z$. Points A, B, C are on the top boundary of the cut, $y = +0$. Points D, E, F are on the bottom boundary of the cut, $y = -0$.

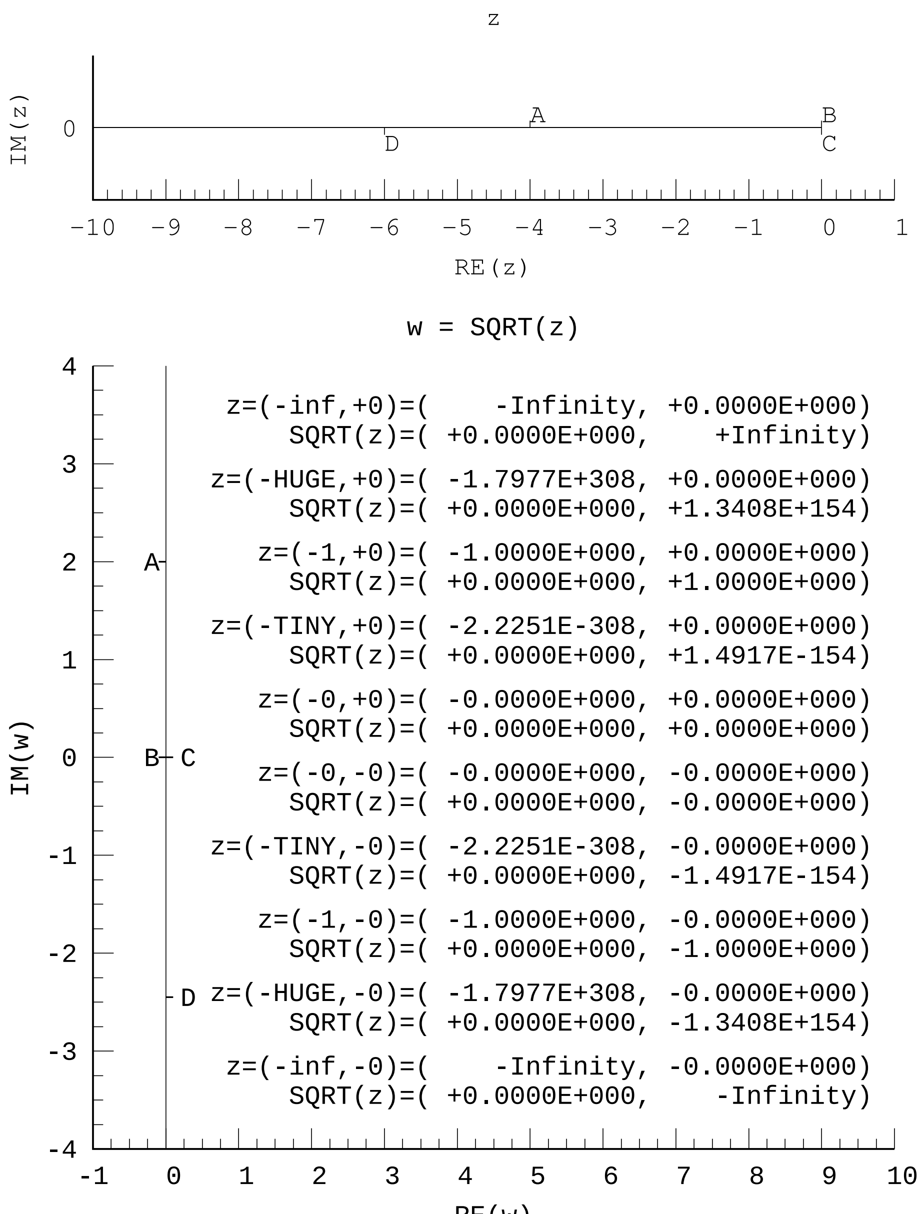

Fig. 2. Map of the branch cut of $w = \sqrt{z}$. Points A and B are on the top boundary of the cut, $y = +0$. Points C and D are on the bottom boundary of the cut, $y = -0$. Points B and C are at $x = -0$.

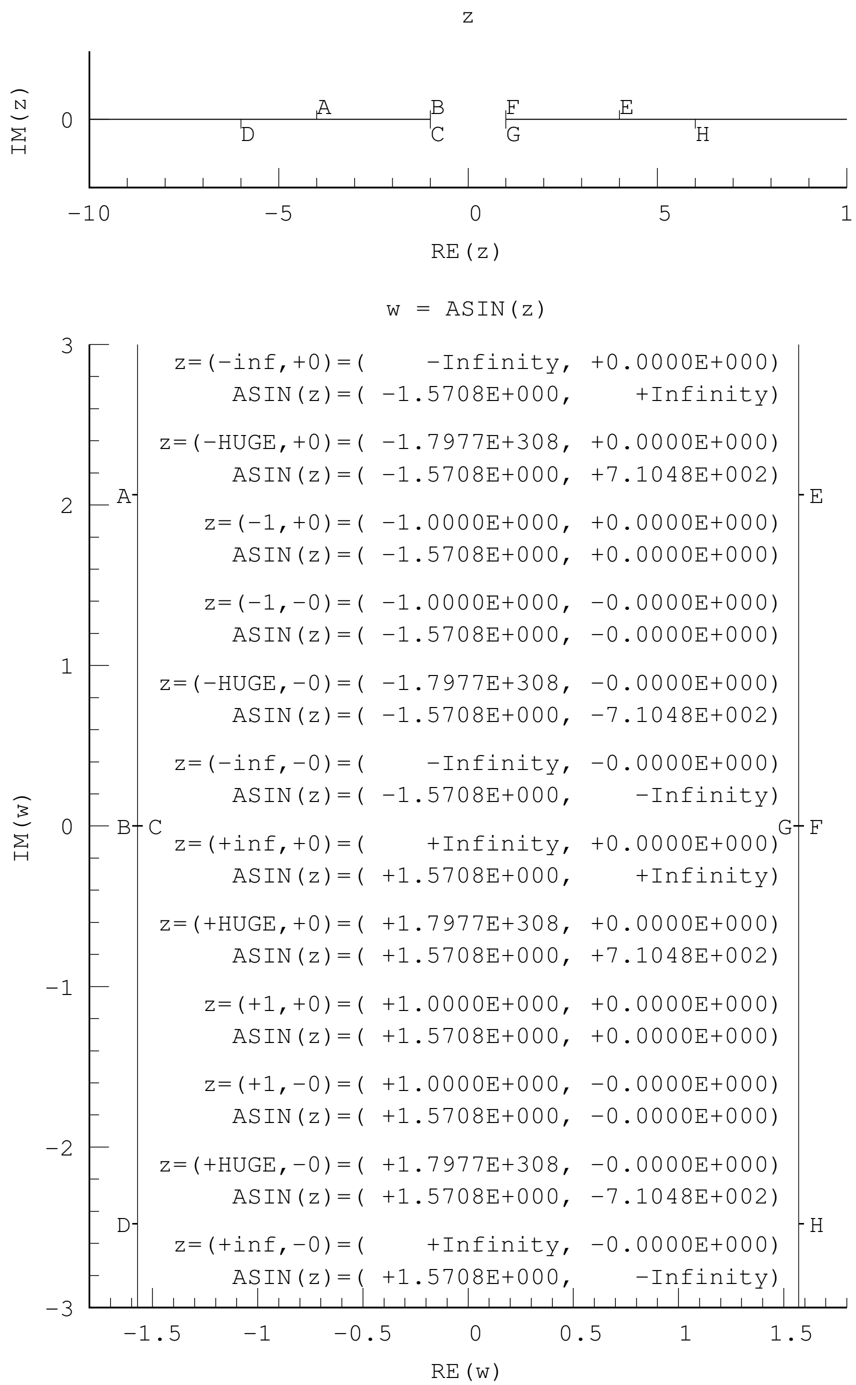

Fig. 3. Map of the branch cut of $w = \arcsin z$. Points A, B, E, F are on the top boundary of the cut, $y = +0$. Points C, D, G, H are on the bottom boundary of the cut, $y = -0$. Points B and C are at $x = -1$. Points F and G are at $x = 1$.

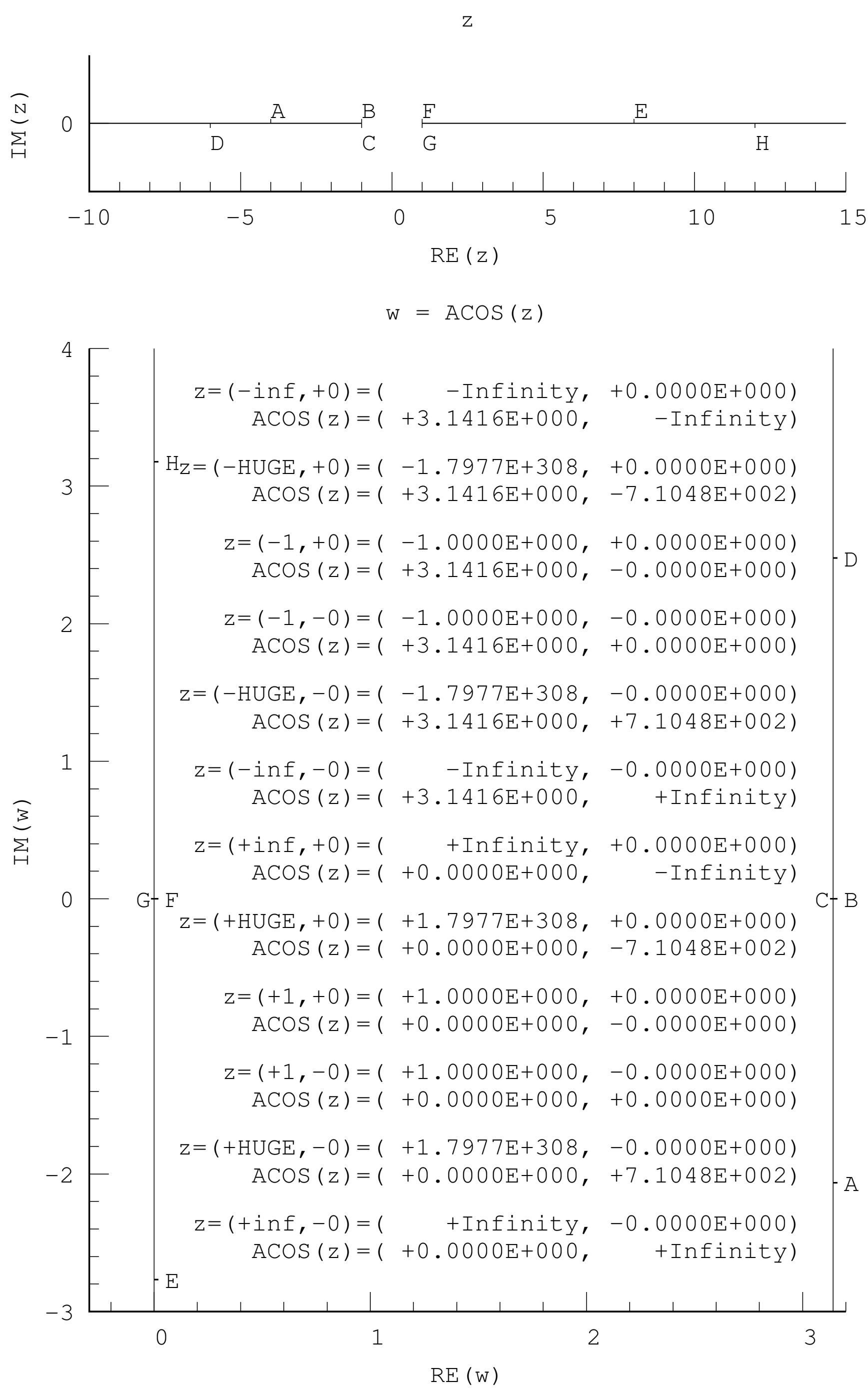

Fig. 4. Map of the branch cut of $w = \arccos z$. Points A, B, E, F are on the top boundary of the cut, $y = +0$. Points C, D, G, H are on the bottom boundary of the cut, $y = -0$. Points B and C are at $x = -1$. Points F and G are at $x = 1$.

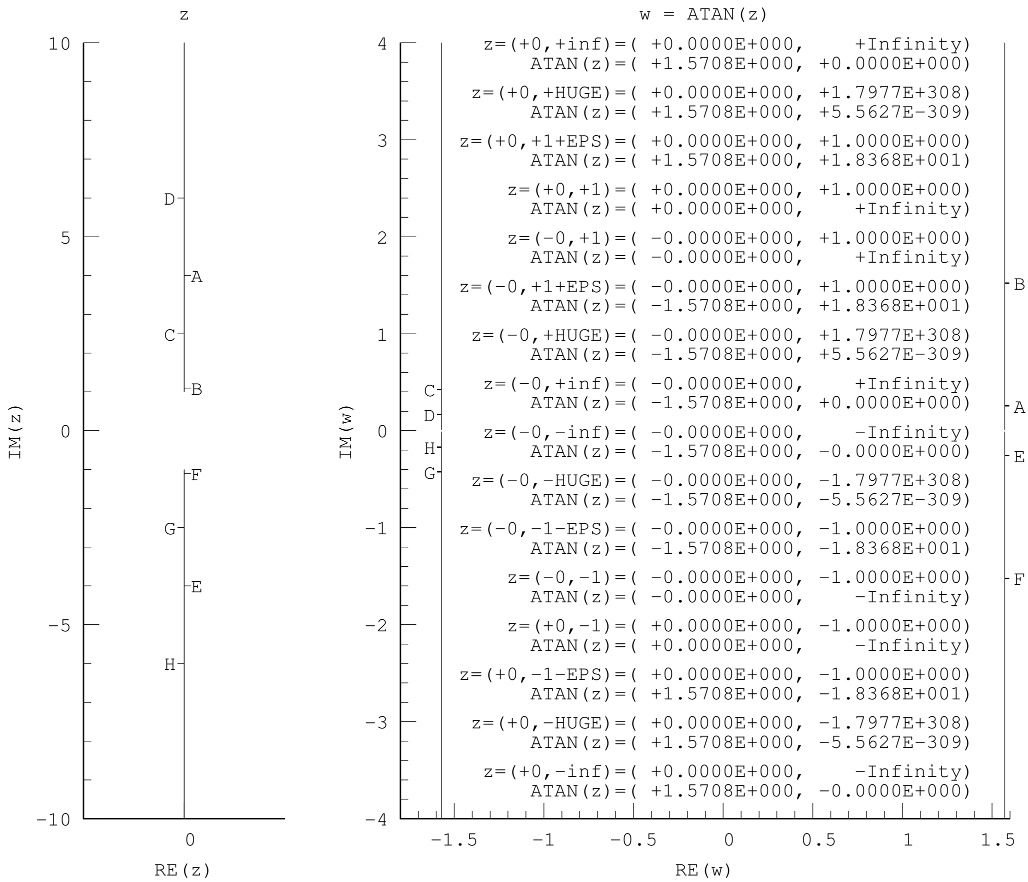

Fig. 5. Map of the branch cut of $w = \arctan z$. Points A, B, E, F are at $x = +0$. Points C, D, G, H are at $x = -0$.

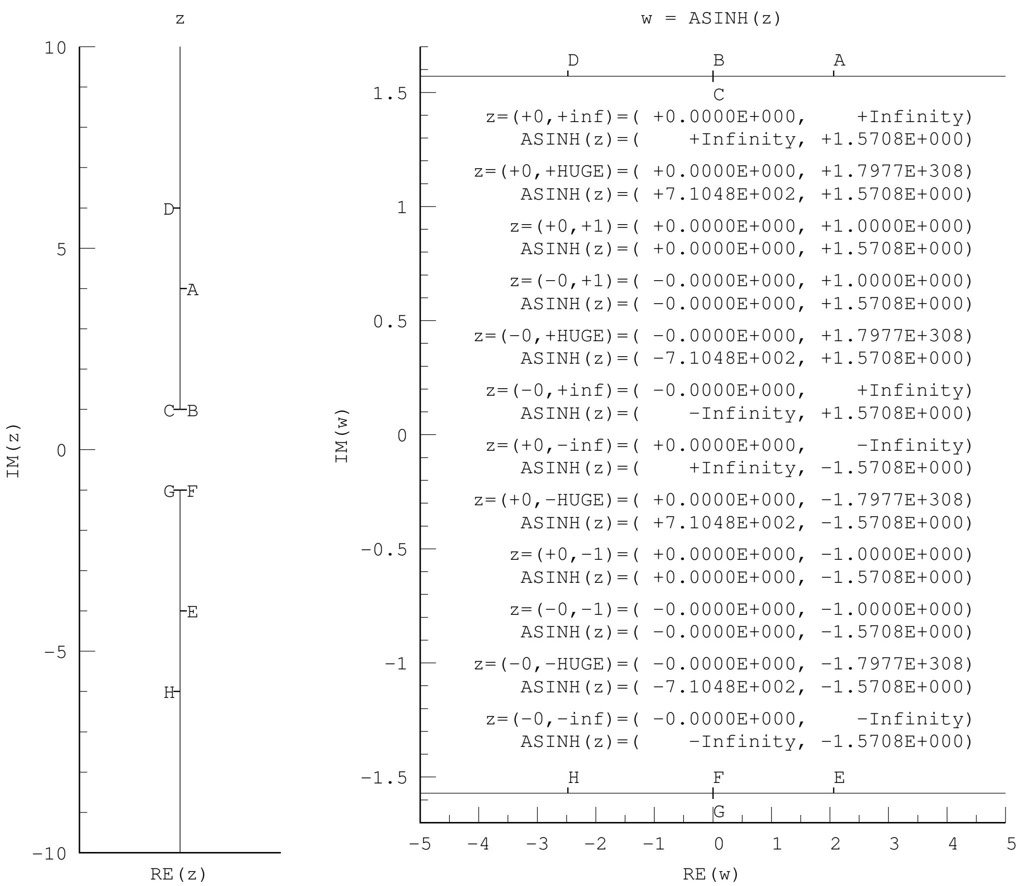

Fig. 6. Map of the branch cut of $w = \operatorname{arcsinh} z$. Points A, B, E, F are at $x = +0$. Points C, D, G, H are at $x = -0$. Points B, C are at $y = 1$. Points F, G are at $y = -1$.

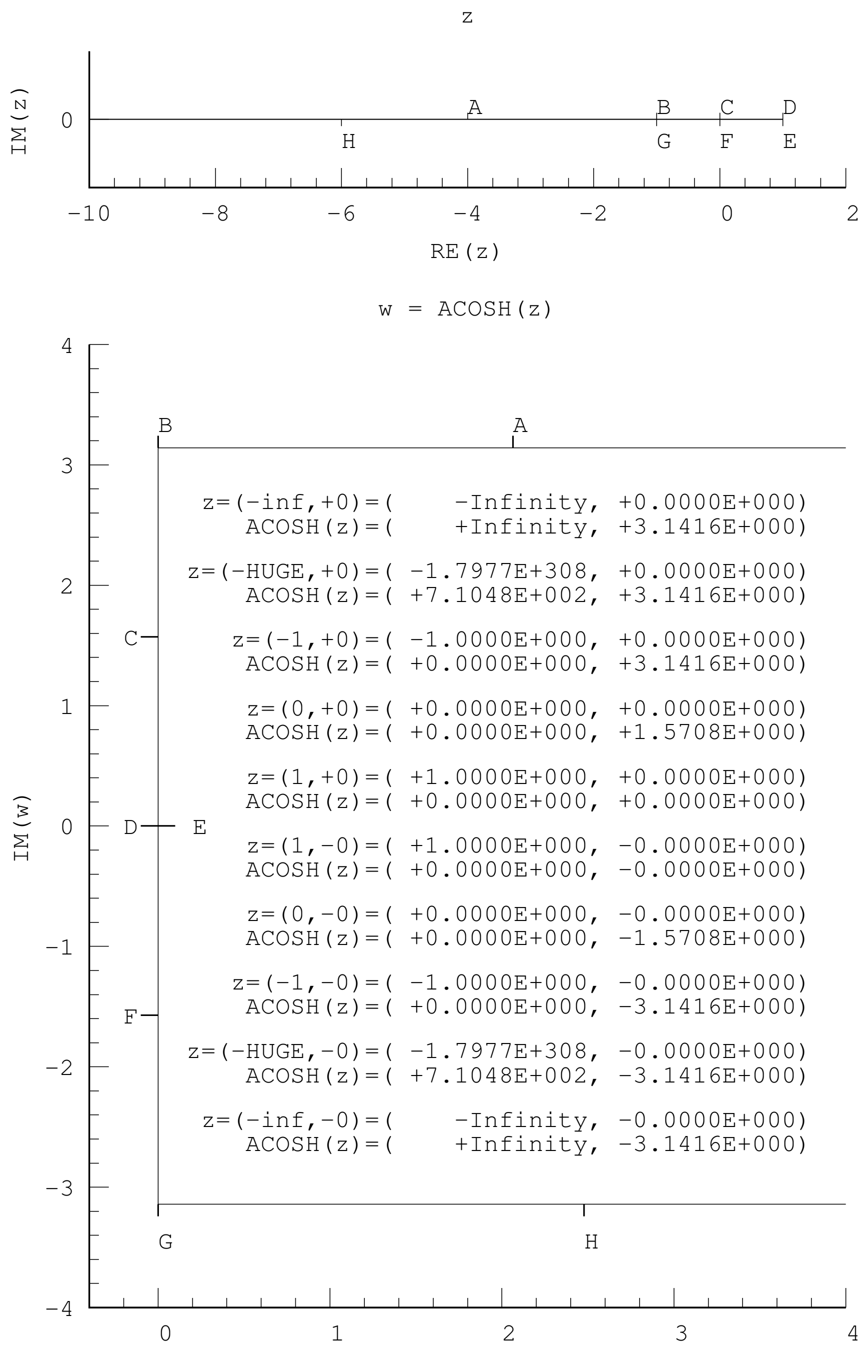

Fig. 7. Map of the branch cut of $w = \operatorname{arccosh} z$. Points A, B, C, D are at $y = +0$. Points E, F, G, H are at $y = -0$. Points D and E are at $x = 1$.

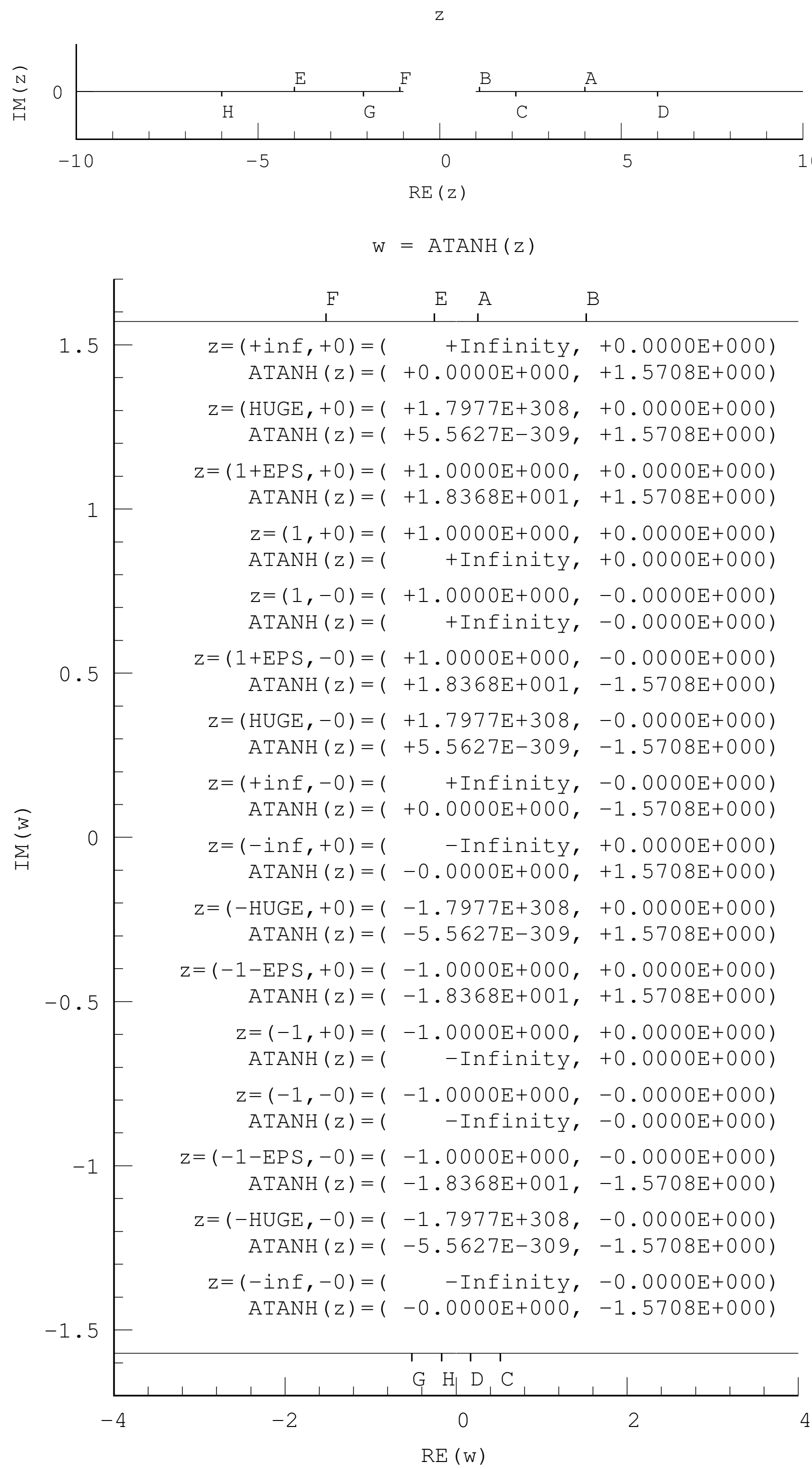

Fig. 8. Map of the branch cut of $w = \operatorname{arctanh} z$. Points A, B, E, F are at $y = +0$. Points C, D, G, H are at $y = -0$.